CLAIM AMENDMENTS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method, comprising:

receiving an optimized library via a network, the optimized library including at least one optimized routine for a processing system; [[and]]

determining whether the optimized library corresponds to an application executing on the processing system in response to a library load request by the application;

loading the optimized library into system memory of the processing system if the optimized library corresponds to the application, wherein the optimized routine is for use by the application to interact with a hardware entity of the processing system; and

utilizing a non-optimized library bound to the application if the optimized library does not correspond to the application, wherein the non-optimized library includes at least one non-optimized routine for use by the application to interact with the hardware entity of the processing system.

~~providing the optimized routine for use by an application executing on the processing system to interact with a hardware entity of the processing system.~~

2. (Original) The method of claim 1 wherein the optimized routine comprises updated code for use by the application to increase interaction efficiency with the hardware entity of the processing system.

3. (Original) The method of claim 1 wherein the receiving the optimized library via the network comprises receiving the optimized library via the network during an operating system ("OS") runtime of the processing system.

4. (Currently Amended) The method of claim 3, further comprising:

receiving an optimization header packet via the network; and

determining that the ~~optimization~~ optimized library is suitable for the processing system based on a module type field within the optimization header packet.

5. (Currently Amended) The method of claim 4 wherein the module type field includes a globally unique identifier ("GUID") for determining that the ~~optimization~~ optimized library is suitable for the processing system.

6. (Original) The method of claim 4, further comprising:

ignoring other optimized libraries broadcast on the network if corresponding other optimization packets are determined to be unsuitable for the processing system based on the module type filed.

7. (Original) The method of claim 3, further comprising:

storing the optimized library to a nonvolatile storage device of the processing system; and

inserting a entry into a pointer table of the processing system, the entry pointing to the optimized library.

8. (Original) The method of claim 7 wherein the pointer table comprises one of a Secondary System Description Table ("SSDT") defined by an Advanced Configuration and Power Interface ("ACPI") and an Extensive Firmware Interface ("EFI") configuration table.

9. (Currently Amended) The method of claim 7 wherein ~~providing the optimized routine for use by the application~~ determining whether the optimized library corresponds to the application, comprises:

executing an optimization extension bound to the application, the optimization extension to request a load of the optimized library; and

querying the pointer table for the entry pointing to the optimized library stored within the nonvolatile storage device[[; and]]

~~loading the optimized library into system memory of the processing system~~.

10. (Currently Amended) The method of claim 9 ~~where providing the optimized routine for use by the application~~ further comprising:

advertising the entry point for the optimized routine of the optimized library to the application, the entry point referencing a location within the system memory of the optimized routine.

11. (Original) The method of claim 9 wherein the optimized library is further loaded into a user mode space of the processing system.

12. (Original) The method of claim 1 wherein the processing system comprises a management module of a rack of blade servers, and further comprising forwarding the optimized library to one or more of the blade servers via an out-of-band channel.

13. (Currently Amended) A machine-accessible medium that provides instructions that, if executed by a machine, will cause the machine to perform operations comprising:

identifying that an optimized library transmitted over a network is intended for the machine, the optimized library including at least one optimized routine for interacting with a hardware entity of the machine;

receiving the optimized library via the network; [[and]]

determining whether the optimized library corresponds to an application executing on the processing system in response to a library load request by the application;

loading the optimized library into system memory of the processing system if the optimized library corresponds to the application;

utilizing a non-optimized library bound to the application if the optimized library does not correspond to the application, wherein the non-optimized library includes at least one non-optimized routine for use by the application to interact with the hardware entity of the processing system; and

advertising one of the non-optimized routine and the optimized routine for use by an application executing in a user mode space of the machine to interact with the hardware entity.

14. (Original) The machine-accessible medium of claim 13 wherein identifying the optimized library, receiving the optimized library, and advertising the optimized library are to be performed during an operating system ("OS") runtime of the machine.

15. (Original) The machine-accessible medium of claim 14 wherein the optimized routine comprises updated code to increase interaction efficiency with the hardware entity of the machine.

16. (Original) The machine-accessible medium of claim 15 wherein the hardware entity comprises a processor of the machine.

17. (Currently Amended) The machine-accessible medium of claim 13 wherein identifying that the optimized library transmitted over the network is intended for the machine further comprises performing operations, including:

receiving an optimization header packet via the network; and

determining that the ~~optimization~~ optimized library is suitable for the machine based on a module type field within the optimization header packet.

18. (Original) The machine-accessible medium of claim 13, further providing instructions that, if executed by the machine, will cause the machine to perform further operations, comprising:

storing the optimized library to a nonvolatile storage device of the machine; and

inserting an entry into a pointer table of the machine, the entry to point to the optimized library.

19. (Original) The machine-accessible medium of claim 18, further providing instructions that, if executed by the machine, will cause the machine to perform further operations, comprising:

executing an optimization extension bound to the application, the optimization extension to request a load of the optimized library;

querying the pointer table for the entry pointing to the optimized library stored within the nonvolatile storage device; and

loading the optimized library into the user mode space of the machine.

20. (Currently Amended) A processing system, comprising:

a processor;

a network link communicatively coupled to the processor; and

a storage device communicatively coupled to the processor, the storage device including instructions which when executed by the processor perform operations, comprising:

monitoring traffic on the network link for an optimized library including at least one optimized routine intended for the processing system;

receiving the optimized library via the network link; [[and]]

determining whether the optimized library corresponds to an application executing on the processing system in response to a library load request by the application;

loading the optimized library into system memory of the processing system if the optimized library corresponds to the application;

utilizing a non-optimized library bound to the application if the optimized library does not correspond to the application, wherein the non-optimized library includes at least one non-optimized routine for use by the application to interact with the hardware entity of the processing system; and

advertising one of the non-optimized routine and the optimized routine to a user mode space of the processing system for use by an application to interact with a hardware entity of the processing system.

21. (Original) The processing system of claim 20 wherein the instructions are to be executed by the processing system during an operating system runtime of the processing system.

22. (Original) The processing system of claim 20 wherein execution of the instructions further performs operations comprising:

parsing an optimization header packet received via the network link; and

recognizing whether the optimized library is intended for the processing system based on a module type field of the of the optimization header packet.

23. (Original) The processing system of claim 20 wherein the application includes an optimization extension to request a load of the optimized library upon execution of the application.

24. (Original) The processing system of claim 20 wherein the hardware entity is the processor.

25. (Original) The processing system of claim 24 wherein the optimized routine comprises updated code for interacting with the processor in a more efficient manner.

26-30 (Canceled)